



Virtual Reality

Frode Opdahl - frode.opdahl@uit.no

What is VR?

Merriam-Webster defines VR as:

"... an artificial world of images and sounds created by a computer that is affected by the actions of a person who is experiencing it."

Basic Concept

- An artificial environment is generated and displayed to the user
- There they are typically able to:
 - Look around
 - Interact
 - Hear sounds
 - Etc.
- External stimulus is robbed for higher immersion



This is VR



But this is also VR...

Virtual Reality isn't a new thing



VR Arcade



Nintendo Virtual Boy (95-96)

But it hasn't had much commercial success...

So why all the hype now?

We are in the future

(Compared to 20 years ago...)

- Computers are powerful enough to render detailed worlds with high frame rates
- Display technology is a lot more advanced
- More advanced motion tracking and sensors
- People are more familiar with technology

Head-Mounted Displays

- User puts on a headset
- An artificial world is rendered on two displays, one for each eye, creating a stereoscopic effect
- When the user moves his head, the artificial world's viewport is moved correspondingly
- This gives the illusion that the user is in the virtual world

VR in 2016

- Head-mounted display (HMD) are all the rage
- They generally have:
 - Head tracking
 - High resolution displays
 - One for each eye
 - Lenses
 - Audio ports / Headphones



The two types of HMDs

- Mobile HMDs
 - Uses your smartphone
 - Google Cardboard
 - Samsung Gear VR
 - Has built in lenses
 - Most popular
- Cheap but not that powerful



Google Daydream

The two types of HMDs

- Designated VR Kits
 - Oculus Rift, HTC Vive, PlayStation VR
 - Requires a powerful computer
 - Advanced displays
 - ~1080x1200 per eye
 - 90-120Hz refresh rate
 - 3D positioning and 3D controllers
- Powerful but expensive!



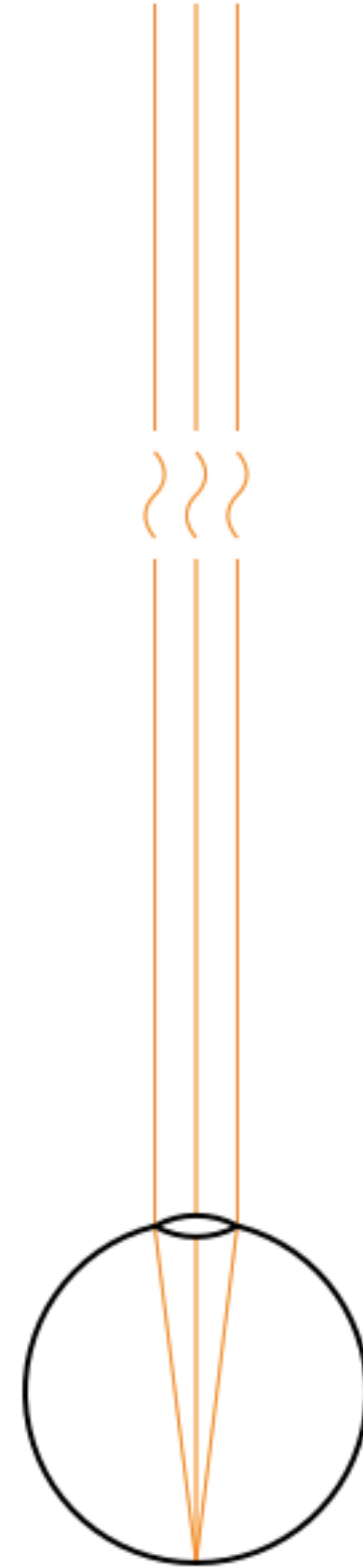
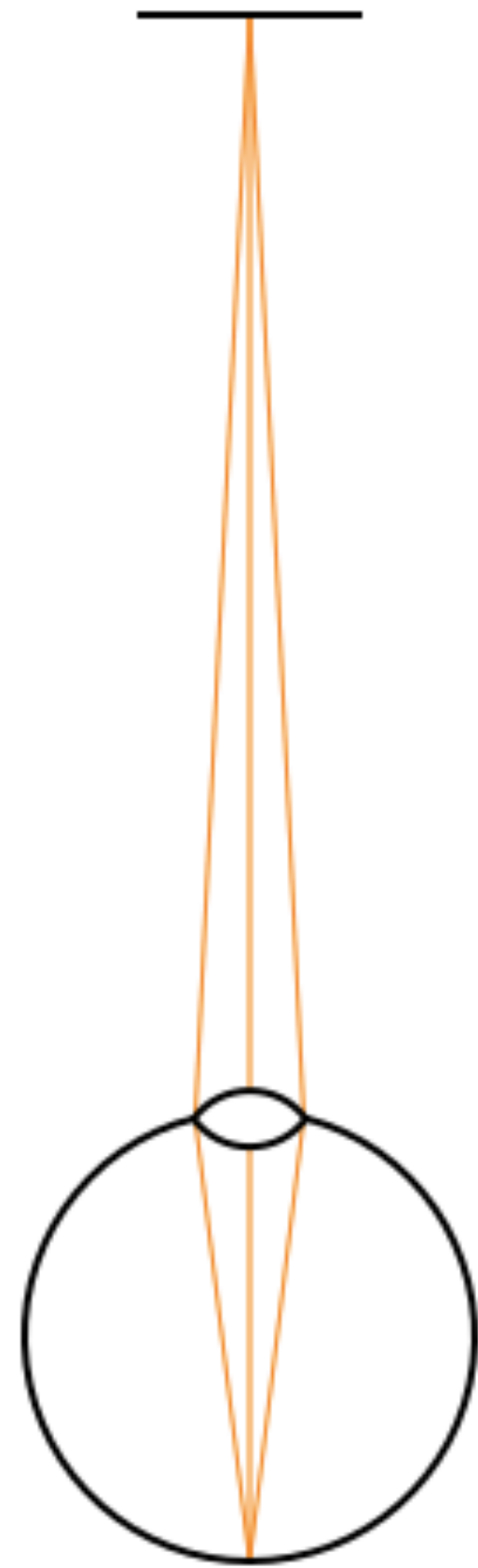
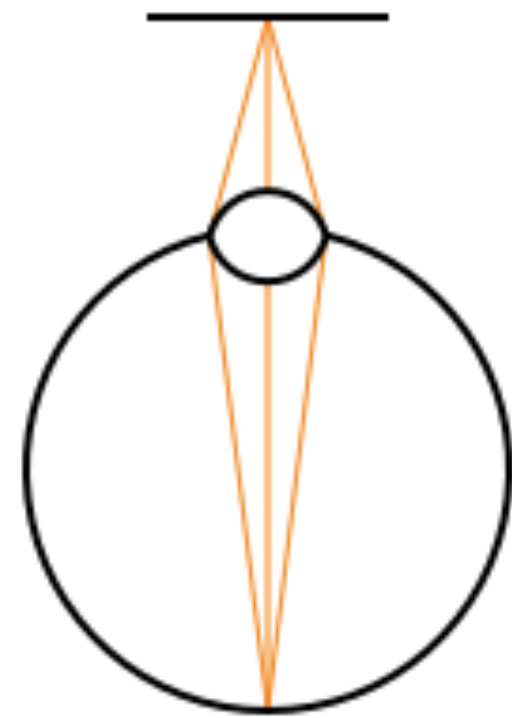
Creating a VR experience
in a headset

Rendering the 3D world

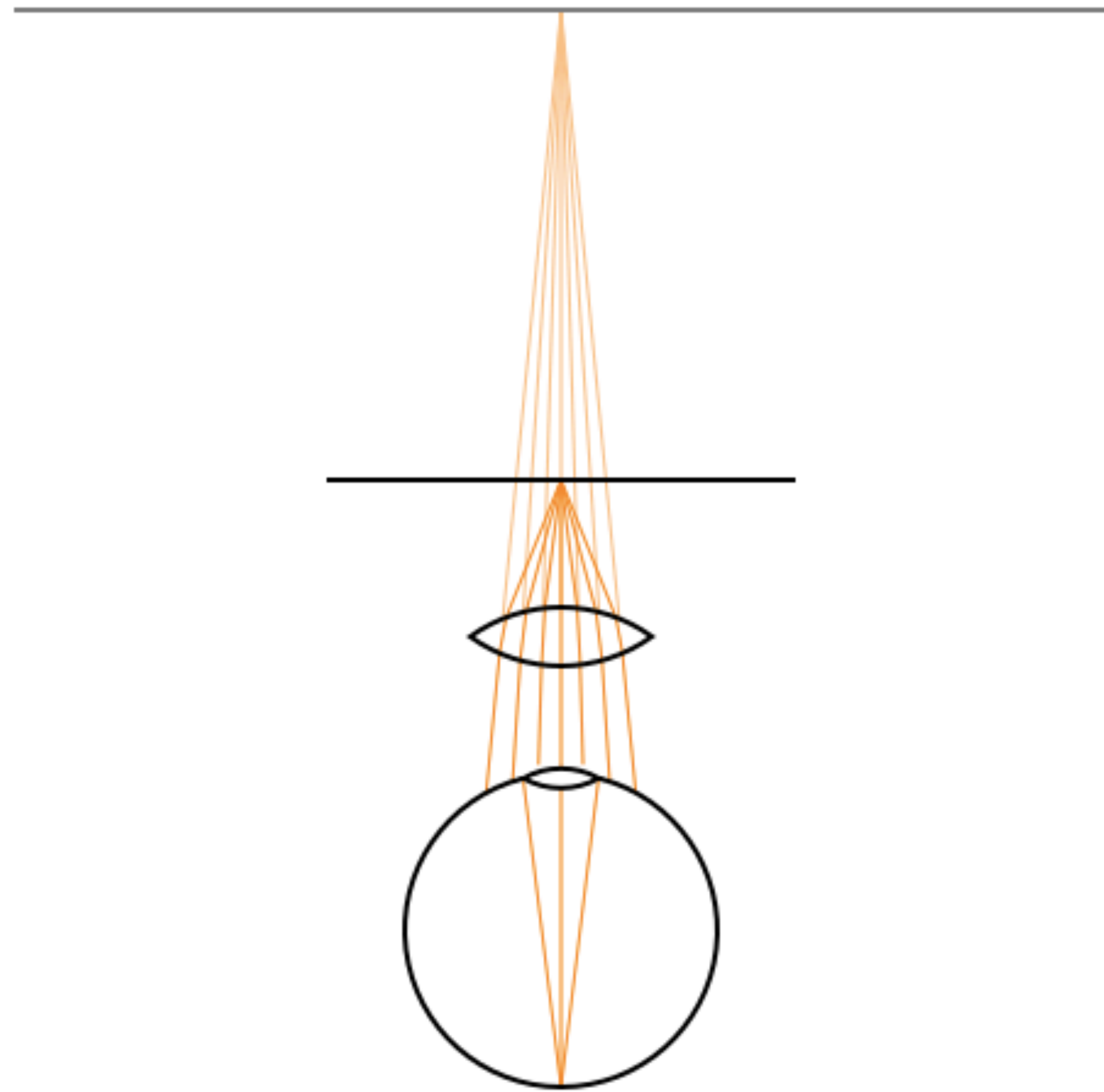
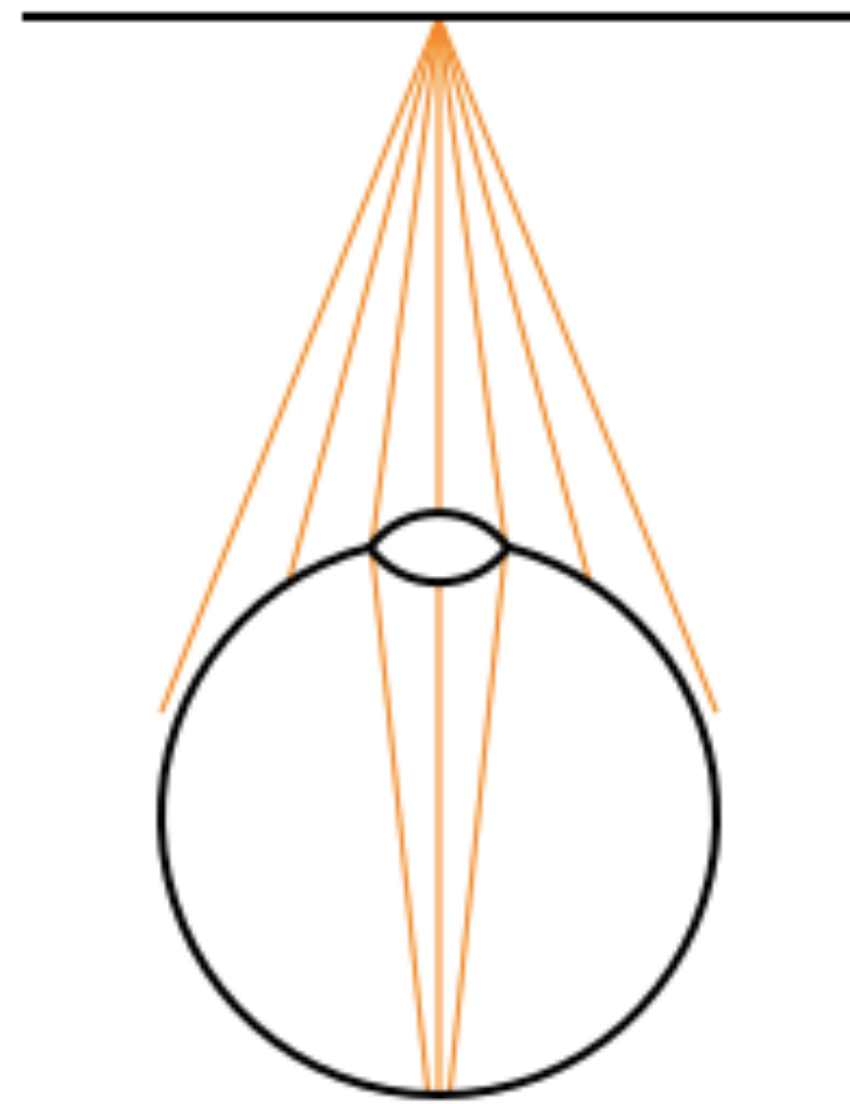
- The same as rendering on a normal monitor
- "All" 3D engines can be made to work with VR
- To achieve depth we want to render two images, one for each eye
- Simplified thinking:
 - Render left eye, move camera ~7cm to the right, render right eye
 - Display it on screen, we have depth inside a 3D world on our headset!

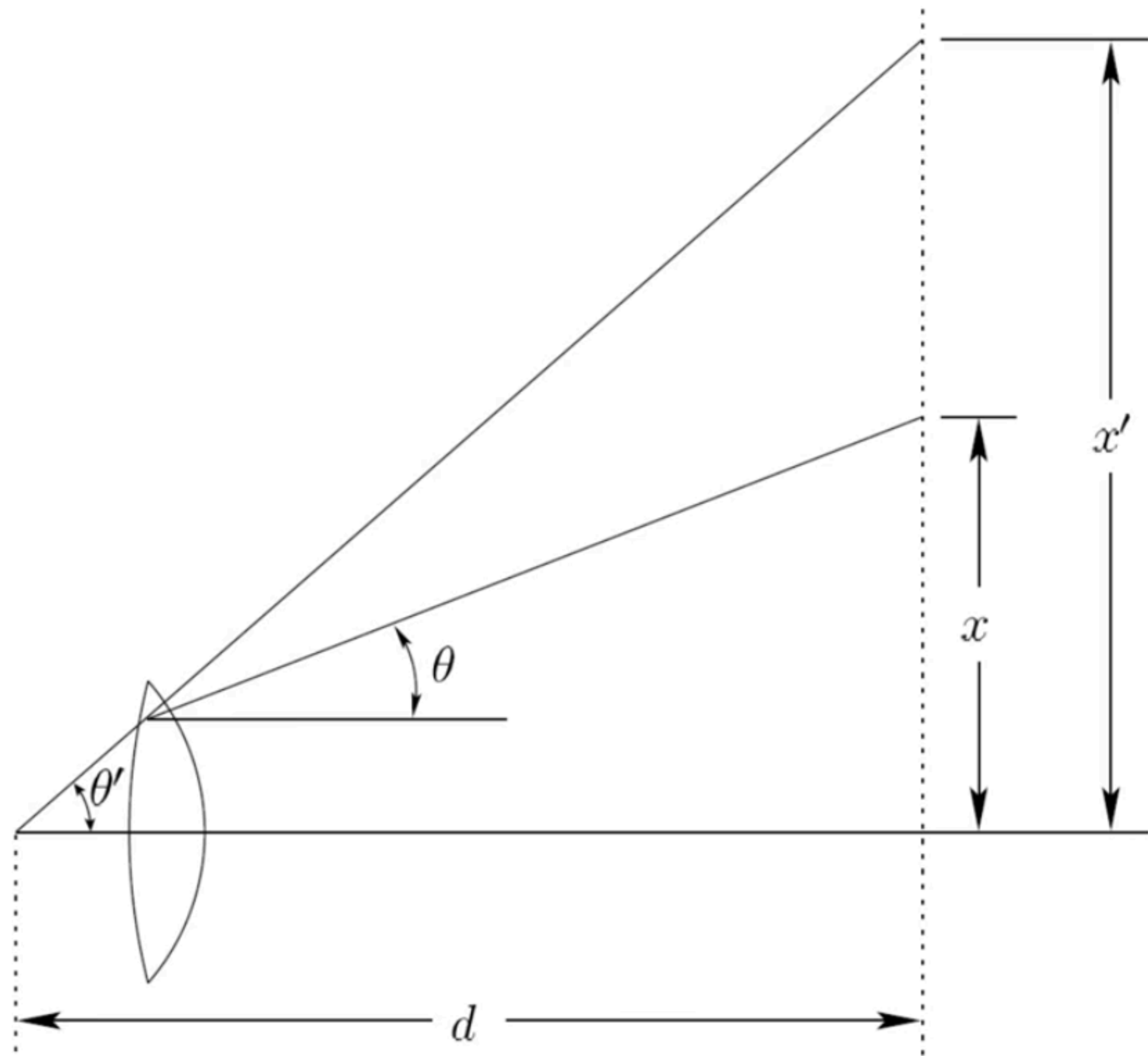
Not quite that simple...

We have to think about how the eye works



An eye focuses on objects at three different distances (near, medium, infinite)

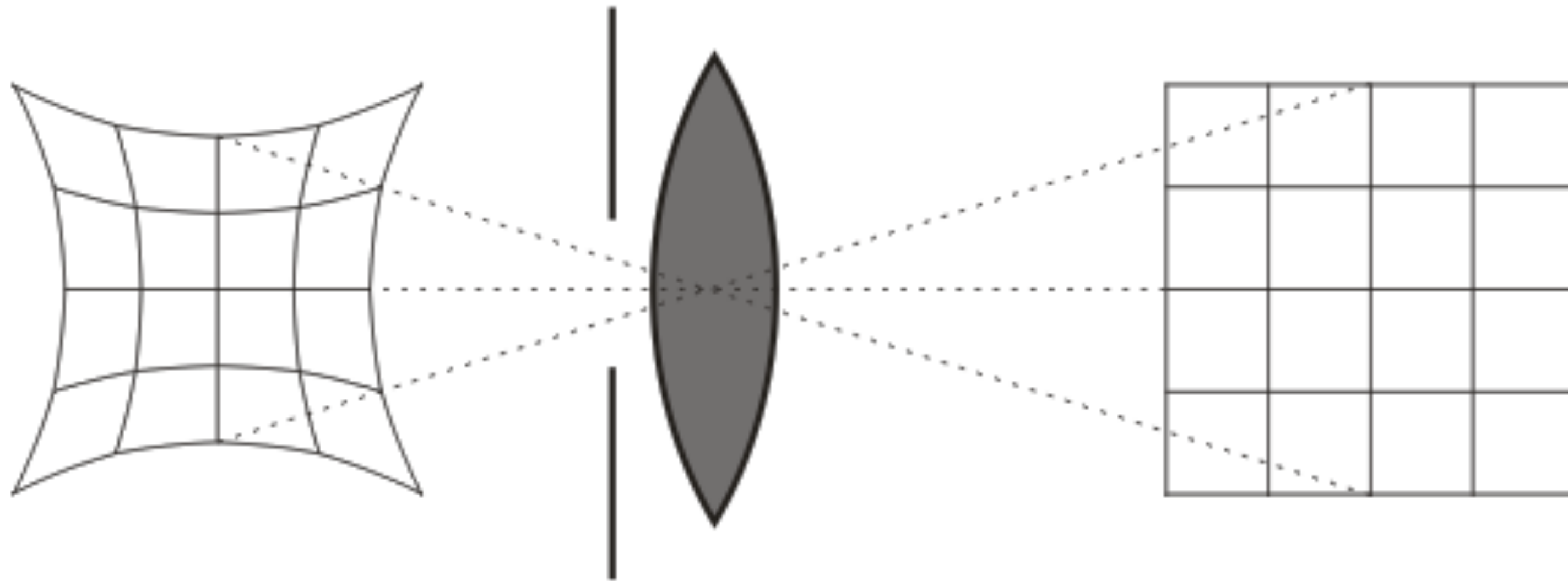




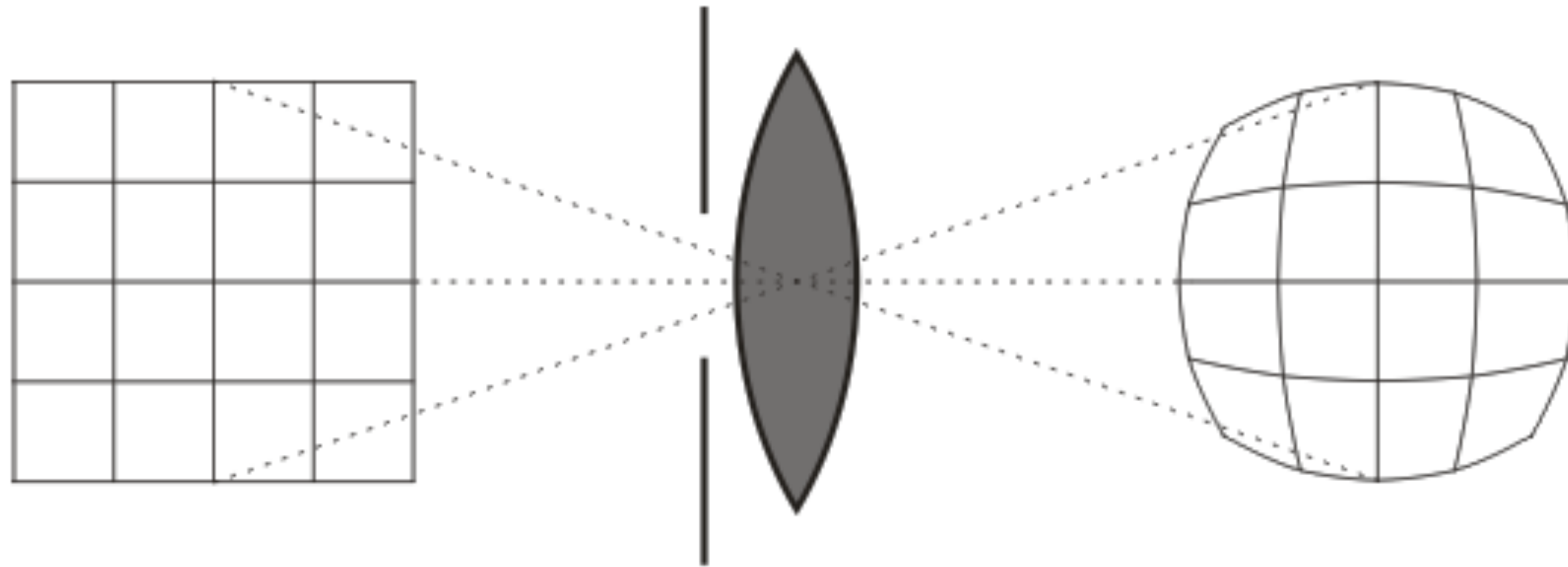
Distorted Image Problem

Perceived Image

Display

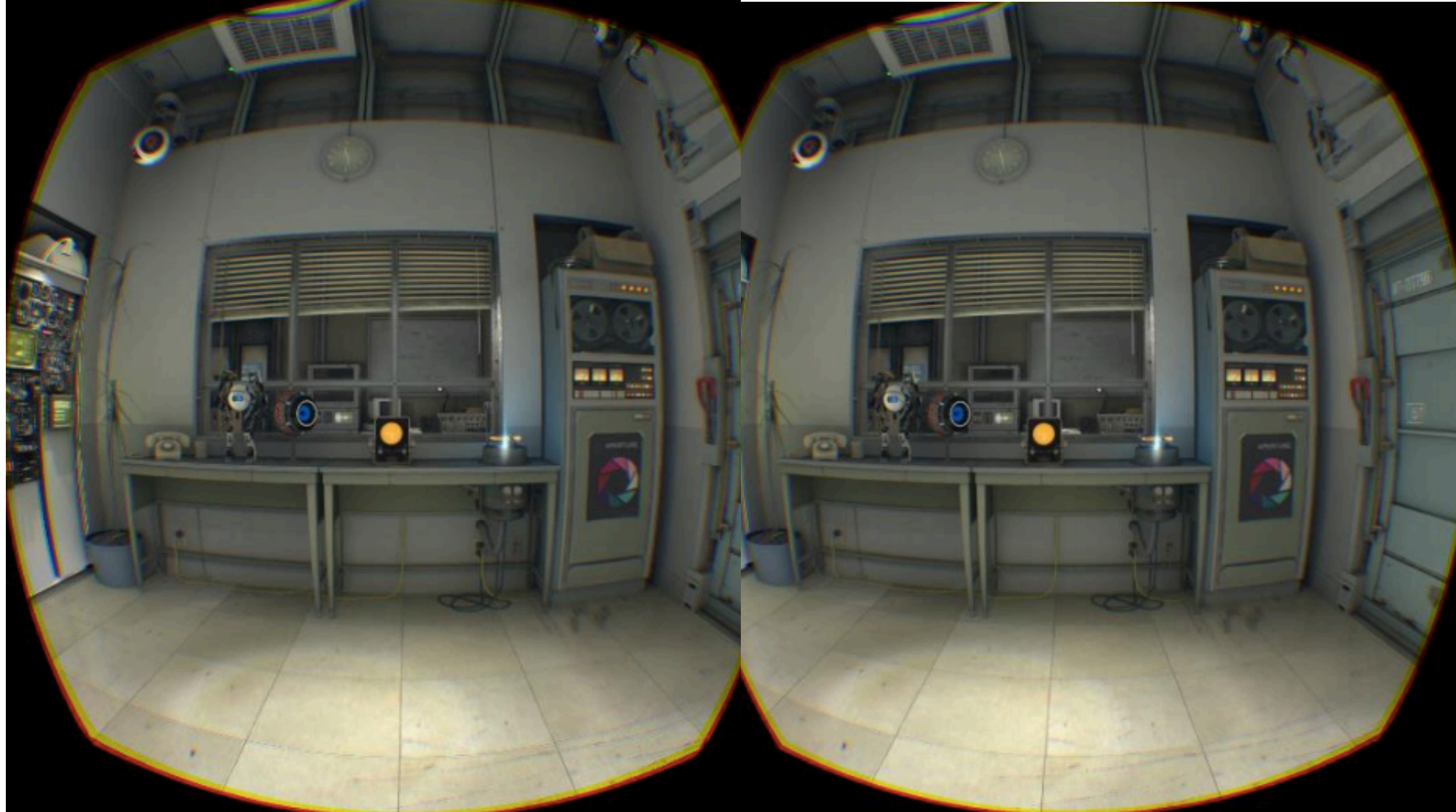


The Solution



Displace the image before it is displayed on the screen

What is rendered on the display



What the user sees



This is expensive...

How much work is required

- 720p @ 30Hz: **27** million pixels/sec
- 1080p @ 60Hz: **124** million pixels/sec
- 30" Monitor 2560x1600 @ 60Hz: **245** million pixels/sec
- 4k Monitor 4096x2160 @ 30Hz: **265** million pixels/sec
- VR 1512x1680x2 @ 90Hz: **457** million pixels/sec

You can do some tricks for better performance

- No need to render black part since the user can't see it
- Render more pixel at the center and less at the edges
- Still only about 30% reduction



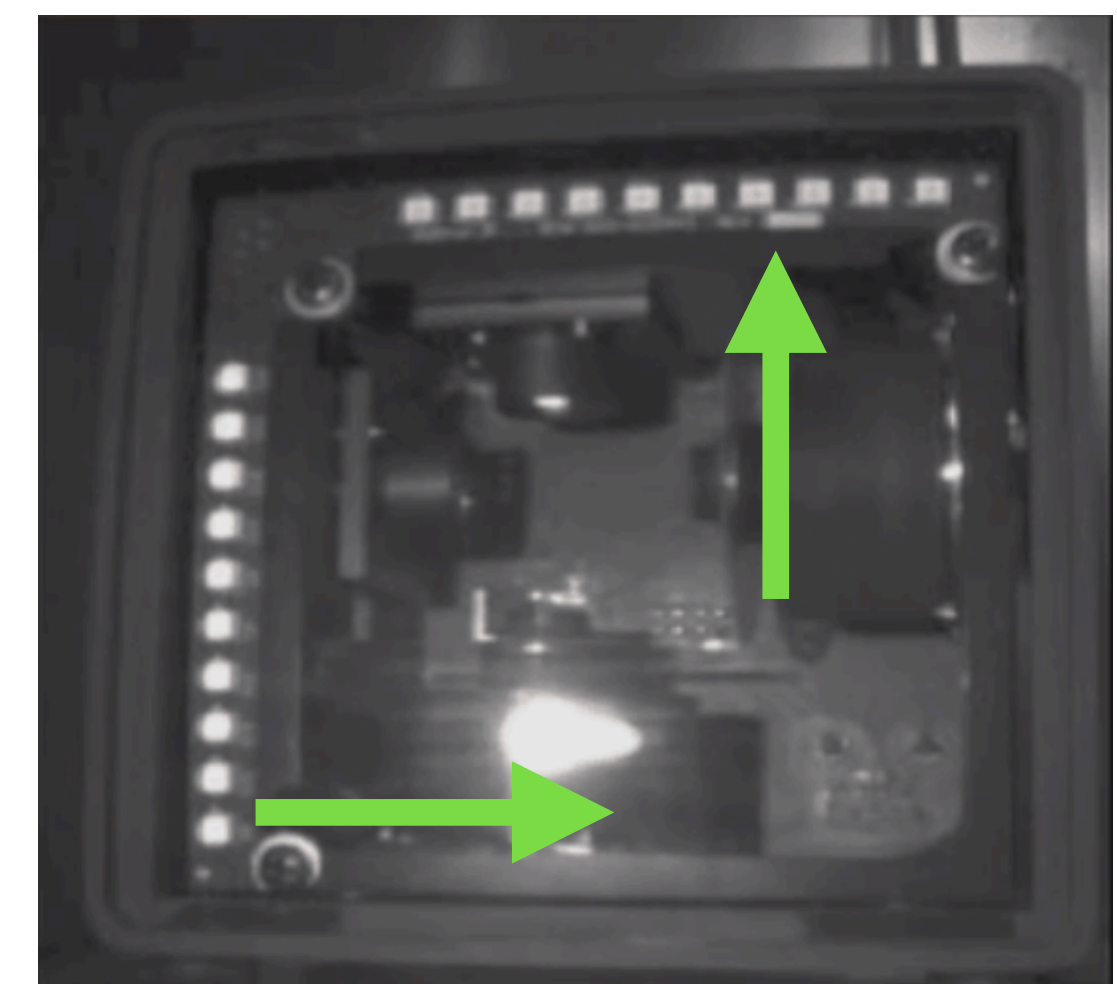
How do you track the user?

Tracking

- Gyroscopes work well, but only tracks rotation
 - Used a lot by mobile HMD
- For designated HMD we want 3D positioning...
 - Want it to accurate (millimeters)
 - Want it low latency
 - There are many different ways to do this

HTC Vive 3D Tracking

- "Lighthouse"
- Cheap and effective tracking
- Requires two light emitters which periodically flashes a light and two lasers pans the room
- Every device looks for the lights emitted from the stations
- Time difference between seeing light flash and seeing laser is used to calculate the position in 3D



Link it together

- Use the tracking data as input for 3D program
- User moves to the left, move the camera to the left etc...
- Same as using a analog stick in a computer game, but with your head

So how do we use this?

Making a VR program

- Incorporate VR into an existing engine
 - Difficult/Complex
 - OpenGL, Direct3D
- Use an engine that already supports VR (let's look at this)
 - Simple if you're familiar with the engine
 - Unity, Unreal 4

Lets look at unity

- Very popular cross-platform game engine
- High level, drag-and-drop interface
- Already has VR support for:
 - Oculus Rift, Google Cardboard, HTC Vive, PlayStation VR, Gear VR, Microsoft Hololens (AR), Google Daydream
- Technical details around VR is already done for you
- More about designing for VR, than implementing VR into your program
- If you know Unity, creating a VR program is no harder than creating any other type of program

Short demo of Unity

So what VR experiences
already exists?

Video Games!



VR Experiences



theBlue, underwater sightseeing



Invasion, short VR film

Medical applications



Helping paralysed people feel their legs

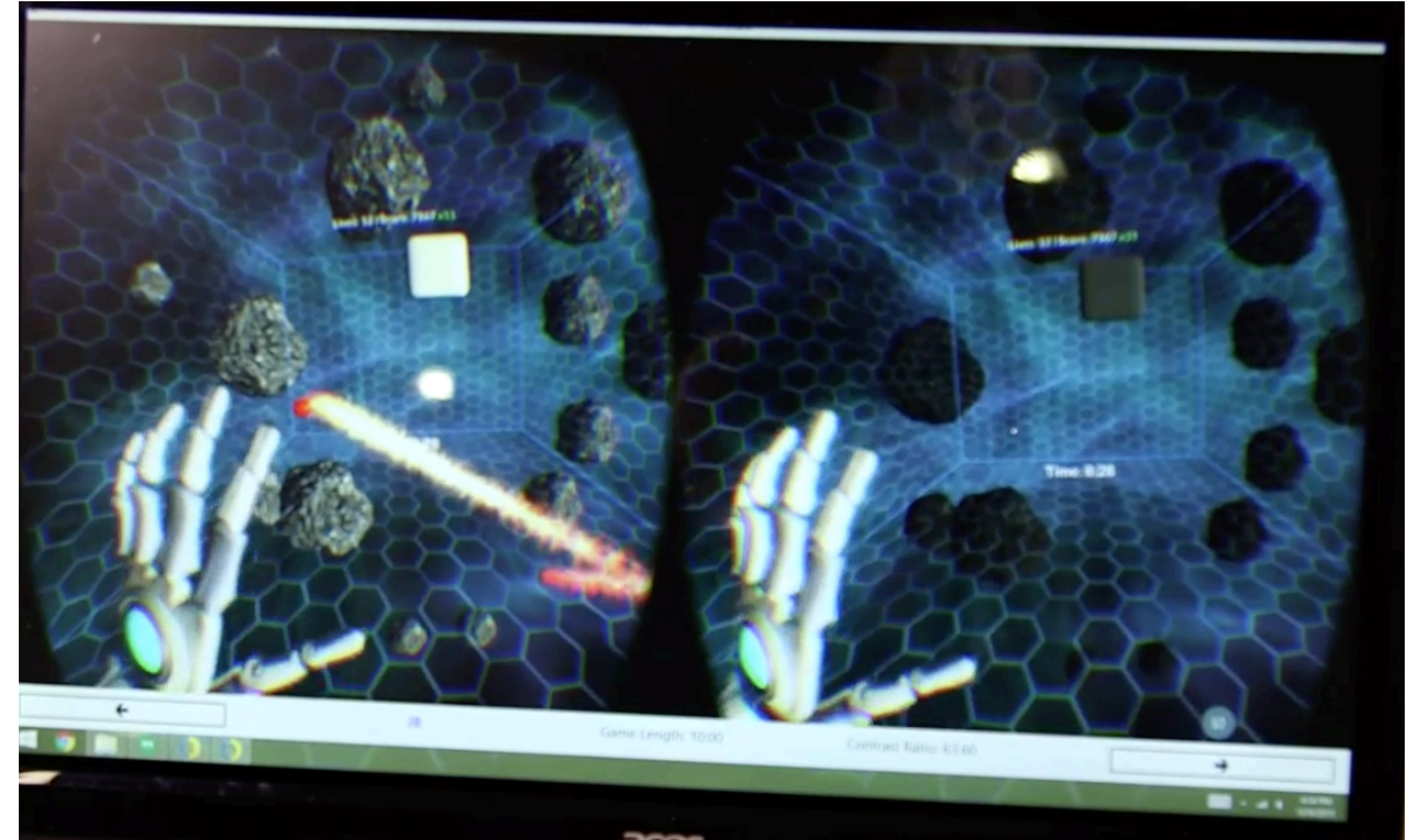


Training surgeons

Medical applications



Relaxing chronic patients



Video game to cure amblyopia
(lazy eye)

Criticism and the future

- "Its an expensive niche"
- Uncertain future, will people buy and use VR products?
- Many big companies are investing into VR
- Predicted to be almost 100 million VR devices by 2020
- Will it go the way of 3D TVs?

Conclusion

- Virtual reality is a rapidly growing industry and its already here
- The basic concept behind the technology is simple
- Developing the technology behind VR is complicated
- Developing programs using VR is fast and easy
- How to use VR properly is harder than programming it